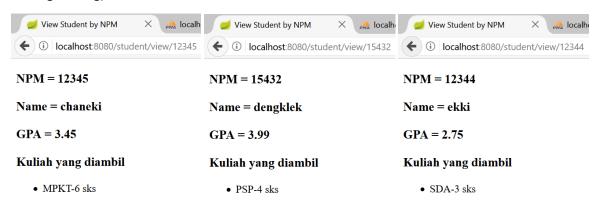
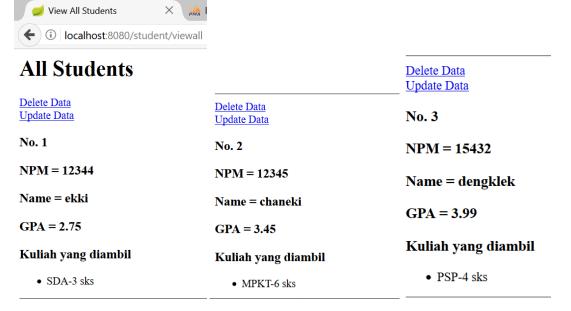
Jalankan program tersebut dan buka http://localhost:8080/student/view/12345 dan http://localhost:8080/student/view/15432 (nomor student disesuaikan dengan database masing-masing).

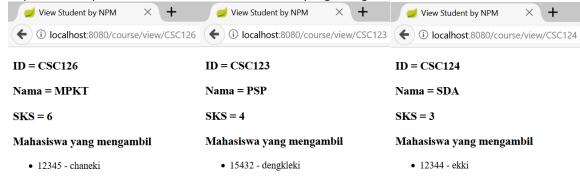


LATIHAN

1. Ubah method selectAllStudents pada kelas **StudentMapper** agar halaman viewall menampilkan semua student beserta daftar kuliah yang diambil.



2. Buatlah view pada halaman http://localhost:8080/course/view/{id} untuk Course sehingga dapat menampilkan data course beserta Student yang mengambil.



Jelaskan apa saja hal yang Anda pelajari dari tutorial ini.

- Pada tutorial kali ini, saya belajar untuk menampilkan data lebih dari satu tabel dengan melakukan join dengan tabel lain

Cantumkan juga penjelasan Anda terhadap hal-hal barikut:

- Method yang Anda ubah pada Latihan Merubah SelectAllStudents , jelaskan

Method yang Anda buat pada Latihan Menambahkan View pada Course, jelaskan
 Menambahkan kode tersebut pad akelas StudentMapper, untuk mengambil atribut pada tabel course, kemudian melakukan join untuk mendapatkan atribut pada tabel student

Menambahkan kode tersebut pada kelas StudentService CourseModel selectCourse(String id);

Menambahkan method tersebut pada kelas StudentServiceDatabase

```
@Override
public CourseModel selectCourse(String id) {
    Log.info("select course with id {}", id);
    return studentMapper.selectCourse(id);
}
```

Menambahkan requestmapping pada kelas StudentController

```
@RequestMapping("/course/view/{id}")
public String cPath(Model model, @PathVariable(value = "id") String id) {
    CourseModel course = studentDAO.selectCourse(id);

    if (course != null) {
        model.addAttribute("course", course);
        return "view2";
    } else {
        model.addAttribute("id", id);
        return "not-found2";
    }
}
```

Membuat file view2.html dan not-found2.html

```
<!DOCTYPE html>
                                                        <!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
                                                       <html xmlns:th="http://www.thymeleaf.org">
<head>
                                                       <head>
<title>View Student by NPM</title>
                                                       <title>Not Found</title>
</head>
                                                        </head>
<body>
                                                       <body>
   <h1>Course not found</h1>
                                                          <h3 th:text="'ID = ' + ${id}">Course ID</h3>
                                                       </html>
   <h3>Mahasiswa yang mengambil</h3>

  th:text="${student.npm} + ' - ' + ${student.name} ">
         NPM - Nama
   </body>
</html>
```